

# Nume proiect

## Introducere

### Prezentarea pe scurt a proiectului:

- **Ce face?** - Proiectul meu este un Sand Drawing Machine, un dispozitiv controlat de Arduino care folosește un magnet mișcat pe două axe (X și Y) pentru a ghida o bilă metalică la suprafața nisipului, desenând modele și forme prestabilite. Utilizatorul poate selecta diferite moduri de desen folosind butoane, iar traiectoriile sunt preluate de pe un card SD. Informațiile despre modul activ sunt afișate pe un display OLED.
- **Care este scopul lui?** - Scopul proiectului este de a crea un sistem automat capabil să deseneze modele geometrice sau artistice în nisip, oferind o combinație de artă și tehnologie. Proiectul urmărește să demonstreze controlul precis al mișcării pe două axe folosind Arduino și să ofere o interfață intuitivă pentru selectarea și afișarea modurilor de desen.
- **Care a fost ideea de la care ați pornit?** - Ideea proiectului a pornit după ce am descoperit pe internet conceptul de sand drawing machines. Mi s-a părut extrem de interesant modul în care o bilă poate fi ghidată de un magnet pentru a desena forme perfecte în nisip. Am vrut să accept provocarea și să văd dacă pot realiza și eu un astfel de sistem folosind componente accesibile și cunoștințele dobândite.
- **De ce credeți că este util pentru alții și pentru voi?** - Consider că proiectul este util pentru că îmbină partea educațională cu cea artistică. Pentru mine, a fost o oportunitate excelentă să învăț mai multe despre controlul motoarelor, comunicarea între module (I2C, SPI) și programarea Arduino. Pentru alții, poate servi ca inspirație pentru proiecte creative, pentru studiu sau chiar ca obiect decorativ interactiv care aduce un plus de originalitate oricărui spațiu.

## Descriere generală

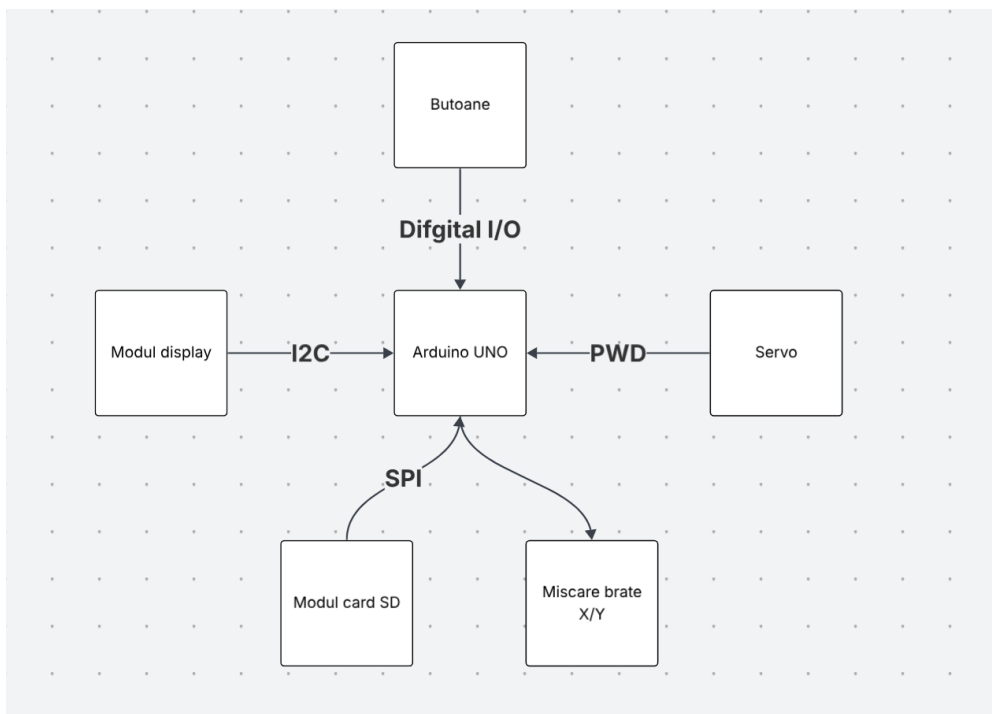
### Descriere module

- **Arduino (Microcontroller)**
  1. Centrul de comandă al întregului sistem.
  2. Primește input de la butoane.
  3. Citește fișiere de pe cardul SD.
  4. Controlează motoarele și afișajul.
- **Butoane (Digital I/O)**
  1. Permite utilizatorului să schimbe modul de desen.
  2. Trimite semnal către Arduino.

- **Modul SD + Card SD (SPI)**
  1. Stochează timestamp-uri pentru desen.
  2. Arduino accesează datele prin SPI.
- **Display OLED / LCD (I2C)**
  1. Afișează modul activ și starea sistemului.
  2. Comunică prin I2C cu Arduino.
- **Driver Motoare (PWM)**
  1. Controlează motoarele pe axele X și Y.
  2. Mișcă magnetul pe sub nisip.
- **Magnet Assembly + Bila metalică**
  1. Magnetul mișcă bila pe nisip pentru a realiza desenul.

## Mod de interacțiune

1. Utilizatorul apasă un buton → Arduino detectează schimbarea.
2. Arduino citește fișierul de pe cardul SD.
3. Arduino trimite comenzi către driver-ele motoarelor.
4. Magnetul se mișcă sub nisip și atrage bila → creează desenul.
5. Display-ul OLED afișează modul activ.



## Hardware Design

## Listă de piese

- Arduino UNO R3
- Driver motoare ULN2003
- 2x motoare stepper 28BYJ-48
- Modul SD Card + card microSD
- LCD 1602 cu I2C
- 1 buton
- Breadboard
- Fire de legătură
- Sursă de alimentare externă (5V - 12V, în funcție de motoare)

## Materialie suplimentare

- Carcasă imprimată 3D sau realizată manual pentru structură
- Placă de nisip
- Nisip fin
- Bilă metalică
- Magneti neodim pentru controlul bilei



## Bill of Materials (BOM)

Nr.	Componentă	Descriere
1	Arduino UNO R3	Microcontroller principal

2	2x 28BYJ-48 + 2x ULN2003 Driver	Motoare pas cu pas + drivere ULN2003
3	LCD 1602 I2C	Display pentru afișare moduri desen
4	Micro SD Card Adapter	Modul scriere fișiere pe Micro SD
5	Buton push	Selectare mod desen
6	Breadboard 800 puncte	Pentru conexiuni temporare
7	Fire (40 bucăți)	Conexiuni între componente
8	Sursă alimentare 5V	Alimentare pentru Arduino
9	Bila metalică + magnet neodim	Componente mecanice pentru desen

## Funcționalitate hardware

Dispozitivul se bazează pe un Arduino UNO, care controlează două motoare pas cu pas 28BYJ-48, fiecare legat la un driver ULN2003. Cele două motoare mișcă un magnet într-un plan XY sub o tavă cu nisip, ghidând o bilă metalică pentru a desena modele.

LCD-ul 1602 comunică cu Arduino prin interfață I2C și afișează modul curent de desen.

Modulul SD este conectat prin interfață SPI și scrie fișiere `.TXT` pe un SD Card log-urile.

Butonul permite utilizatorului să schimbe între diverse moduri de desen.

## Pini folosiți (Arduino UNO)

Componentă	Pin Arduino	Tip conexiune	Detalii tehnice
Stepper 1	D8-D11	PWM	Control IN1-IN4 prin ULN2003
Stepper 2	D4-D7	PWM	Control IN1-IN4 prin ULN2003
LCD I2C	A4 (SDA), A5 (SCL)	I2C	Comunicare I2C standard la 0x27
CH376S	D0 (RX), D1 (TX)	SPI	Comunicarea SPI pentru citirea SD Card
Buton	D12	Digital Input	



## Software Design

### Descrierea codului aplicației (firmware)

- **Mediu de dezvoltare:** Arduino IDE, programarea s-a realizat în limbajul C/C++ specific platformei Arduino.
- **Librării și surse 3rd-party:**
  - `Stepper.h` - folosită pentru controlul motoarelor pas cu pas 28BYJ-48 prin driverele ULN2003.
  - `LiquidCrystal\_I2C.h` - pentru controlul unui ecran LCD 1602 prin interfața I2C (cu backlight).
  - `SPI.h` și `SD.h` - pentru inițializarea și operarea modulului de card microSD pe magistrala SPI.

**• Algoritmi și structuri implementate:**

- Mișcare sincronă a celor două motoare folosind funcția personalizată `stepTogether`, care calculează pașii proporțional în funcție de traiectoria dorită.
- Funcții abstracte `stepX` și `stepY` pentru a naviga în planul X/Y, care țin cont de limitele de mișcare și actualizează poziția curentă.
- Implementare a două moduri de desen: **zigzag** și **spirală**, selectabile cu un buton și afișate pe ecranul LCD.
- Algoritm PWM software (cu `digitalWrite` și `delayMicroseconds`) pentru controlul vizual pe un LED conectat la pin analogic.
- Comenzi seriale interpretate în `loop()` pentru a permite controlul manual al brațului desenator din tastatură (WASD și alte shortcut-uri).

**• Funcții implementate principale:**

- `drawZigZag(int size)` – desenează modelul zigzag pe întreaga suprafață.
- `drawSquareSpiral(int size)` – desenează o spirală pătrată cu pasul specificat.
- `stepX(int steps)` și `stepY(int steps)` – mută bila în plan pe axele X/Y.
- `center()` și `goHome()` – re poziționează bila în centrul sau la originea suprafeței.
- `logFile = SD.open(...)` – scrie un log cu tipul desenului și timpul de execuție în milisecunde.

---

## Rezultate Obținute

- [Desenare \(Sand Drawing\)](#)
- [Selectare moduri](#)
- [Control PWM pe LED](#)

## Concluzii

Proiectul „Sand Drawing Machine” a fost o oportunitate excelentă de a combina partea de hardware cu cea software într-o aplicație creativă și interactivă. Utilizând un Arduino Uno, două motoare pas cu pas 28BYJ-48 și un ecran LCD cu interfață I2C, am reușit să construiesc un dispozitiv care poate desena modele precum spirale și zigzaguri într-un strat de nisip, cu ajutorul unei bile metalice ghidate magnetic.

Pe parcursul dezvoltării am întâlnit provocări legate de sincronizarea motoarelor, controlul precis al poziției, citirea de pe cardul SD și afișarea informațiilor pe LCD, dar toate acestea au fost depășite prin testare și implementarea unor soluții personalizate.

Funcționalitățile software, precum controlul PWM al LED-ului, salvarea logurilor pe cardul SD și comutarea modurilor prin buton, au adus un plus de interactivitate proiectului. Am învățat importanța gestionării resurselor hardware limitate ale microcontrollerului, precum și avantajele modularizării codului în funcții clare.

Consider că proiectul poate fi extins în viitor prin adăugarea unor senzori pentru auto-calibrare, integrarea unei aplicații mobile pentru control de la distanță sau adăugarea unor modele mai complexe citite de pe fișiere externe.

**În concluzie**, Sand Drawing Machine demonstrează cu succes cum pot fi combinate cunoștințele de electronică, programare și design pentru a crea un sistem funcțional, estetic și educativ.

## Download

[cod\\_sand\\_drawing.zip](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/atoader/alexandru.stefan03>



Last update: **2025/05/28 00:06**